AMENDMENTS TO THE SPECIFICATION

Please correct the paragraph beginning at page 2, line 10 as follows:

All such solutions share a substantial number of components, be they implemented in software or hardware. Because all the cryptographic packages share so many components in common, they are implemented and offered as a single package from which one chooses which components, or level of cryptography, to enable or use. In a software implementation, the desired cryptographic support may be compiled~~ in~~ during a system build or selectively invoked using various system initialization or configuration parameters. In a hardware implementation ~~the~~ a chip is typically enabled for a certain type of cryptographic support during system initialization using system boot parameters.

Please amend the paragraph beginning at page 3, line 6 as follows:

Software ~~implementation~~ implementations have the advantage that they can be custom built. However, that solution also has overwhelmingly severe disadvantages. First, all known implementations of cryptographic features are notoriously compute intensive. If the cryptography in use is software based, and the system is to be used for anything other than just encryption and decryption, system performance will always degrade to a very noticeable degree. This is unacceptable, so software solutions are not as common except in dedicated security systems. In addition to poor performance, using different builds for different cryptographic support results in severe logistical problems as the number of releases grows. Thus, the only pragmatic solution is to use some kind of system initialization parameter.

Please amend the paragraph beginning at page 8, line 19 as follows:

The boot strap code will typically be kept in ROM 110. When the CPU is ready to start initializing the system, ~~it execute~~ the code executes a jump instruction to a specified location in ROM 100 and ~~start~~ starts executing the instructions found there. Since this code is executed before any other code, the ROM must be accessible to and addressable by the CPU before other devices that need initialization. At some point in the instruction stream, the boot strap code will be ready to configure the cryptographic capabilities of the system. Typically this will involve a set of instructions that will enable and disable certain pins on crypto chip 112. In order to carry out the configuration, the crypto init data 106 must be decrypted.

Please amend the paragraph beginning at page 12, line as follows:

The action taken in box 204 is to store the encrypted data (block, string, token, etc.) in non-volatile memory available to the CPU on the CPU's local bus. ~~This~~The encrypted data ~~will~~ could typically be in either system configuration and initialization ROM ~~but could be~~ or in the NVRAM which also holds the boot strap code, as discussed above. After storing the encrypted data, box 204 is left and box 206 entered.

Please amend the paragraph beginning at page 12, line 16 as follows:

Box 206 represents the first box whose actions will occur during system boot. Boxes 200 through 204 are those actions taken to prepare the system so that when ~~it~~ the system boots, ~~it~~ the system will boot with the intended configuration; boxes 206 through 214 are those actions taken every time the system is booted. By the time the actions in box 206 are started, the CPU will have finished its own internal initializations, including power-on self test checks, and will be starting the process of initializing devices found on ~~its~~ local bus of the CPU. To ~~do this~~ perform initialization, the CPU will have started executing boot strap code out of NVRAM, where the boot strap code is stored. At a certain point during the execution of the boot strap code, ~~it will be time to initialize~~ the crypto capabilities of the system must be configured to the desired level. When ~~that point~~ configuration of the crypto capabilities in the code is reached, the first step will be to fully enable the crypto chip, and using the MAC address, generate the key pairs. Using the private key, the encrypted token is decrypted. As will be remembered from the description given above, when using the word "token" any amount of encrypted data is meant – the actual amount of encrypted data to be decrypted will be determined based on considerations including, but not limited to, the highest level of encryption available on this chip or on this system, the algorithms used in that level of encryption, and the type of checking to be done (i.e., the present invention may be used with authentication mechanisms as well as decryption mechanisms).

Please amend the paragraph beginning at page 13, line 14 as follows:

After decrypting the encrypted token, box 208 is left and decision diamond 210 entered. The question to be answered at this point is if the decrypted token has the format (or value or other recognizable measure) expected. If ~~it~~ the token does not, then the "NO" exit is taken and box 214 entered. The actions taken in box 214 are to disable the crypto chip entirely and set a system warning that security has been compromised.